

# Eyetracker를 활용한 아이트래킹 키오스크 시스템

노현수, 김정재, 원종운, 임희호, 이지윤, 정순호  
 부경대학교 컴퓨터공학과  
 e-mail : mertiz36@naver.com

## Eye Tracking Kiosk System using Eyetracker

Hyun-Soo Noh, Jung-Jae Kim, Jong-Un Won,  
 Hee-Ho Lim, Ji-Yoon Li, Soon-Ho Jung  
 Computer Engineering, Pukyong National University

### 요 약

키오스크의 카메라에 촬영된 이미지를 바탕으로 사용자를 인식하여 사용자의 눈 이동을 탐지한다. 사용자가 현재 바라보고 있는 화면의 위치를 선택할 수 있도록 하여, 키오스크에 접촉하지 않음으로써 위생문제를 해결하고 직관적인 서비스를 제공한다.

## 1. 서론

COVID-19 바이러스로 인해 비대면으로 주문할 수 있는 키오스크가 이용률이 높아지고 있으며 [그림 1]을 통해 그 현황을 확인할 수 있다. 그러나 키오스크 화면에 접촉함으로써 또 다른 위생 문제를 일으킬 수 있다는 문제인식을 하게 되었다.



[그림 1] 2018년 ~ 2020년 점포 내 키오스크 사용여부

이 논문에서는 영상 처리 기반 객체 탐지 및 인식 기술을 사용하여 사용자의 얼굴과 시선을 인식함으로써 간단하게 조작할 수 있는 키오스크 시스템을 개발하고자 한다. Harr Casade 알고리즘과 eye-tracking 기술을 융합하여 그 기능을 실험하고 결과를 도출하여 기존의 키오스크와 비교했을 때 더욱 뛰어난 위생효과를 기대한다.

## 2. 관련 연구

키오스크 시스템이 동작하기 위해서 먼저 사용자의 얼굴을 인식한다. OpenCV, 머신러닝을 기반으로 한 오브젝트 검출 알고리즘인 Haar Cascade를 사용하여 이미지에서 사람의 얼굴을 검출한다. 이미지를 스캔하여 적분 이미지로 생성한 다음, Haar 특징을 사용하여 사람의 얼굴을 검출하기 위한 의미있는 영역을 추출하는 방식을 통해 CPU 기준

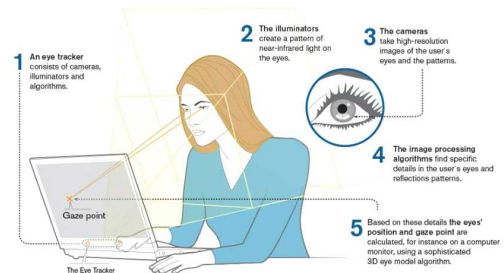
0.159s의 인식속도와 93% 이상의 인식률을 보여준다.[4]

Haar Cascade 알고리즘은 [표 1]과 같다.

Haar Feature Selection	이미지에서 하르 특징 선택
Creating Integral Images	계산속도 향상을 위한 적분 이미지 생성
Adaboost Training	Adaboost Trainig을 통한 이미지 내 의미 있는 특징을 검출
Cascading Classifiers	계산량 감소를 위한 분류

[표 1] Haar Cascade 알고리즘

정밀한 eye-tracking 기능을 구현하기 위해 tobii사의 eye-tracking를 사용한다. 근적외선 LED에 의해 각막에 생성된 반사패턴을 기반으로 각막반사법(PCCR)에 개량한 기술을 구현하며, 빛의 반사점과 기타 기하학적 특징을 바탕으로 사용자의 시선을 추출한다.[5] Eye-Tracking의 원리는 [그림 2]와 같다.



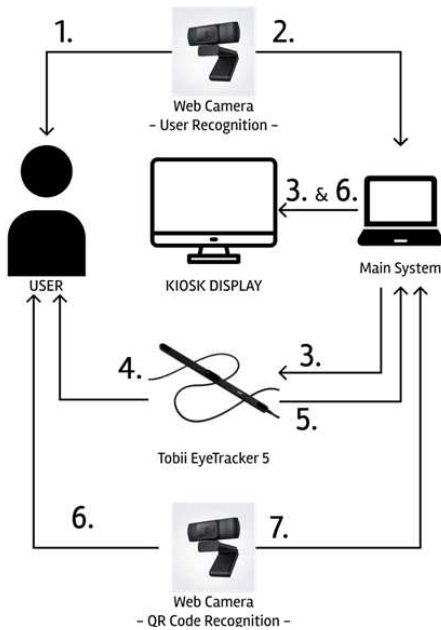
[그림 2] Eye tracking 원리

### 3. 아이트래킹 키오스크

본 시스템에서는 Haar Cascade Detection을 통한 사용자 얼굴 인식, tobii Eye-Tracker, PyQt 기반의 키오스크 UI를 이용하여 구현한다. 시선추적을 통해 키오스크 화면에 터치하지 않고 눈의 움직임만으로 사용자의 요구를 인지하는 것으로 키오스크 서비스를 제공한다. Eye-Tracking 키오스크 시스템은 [그림 3]과 같다.



[그림 3] Eye-Tracking 키오스크 시스템

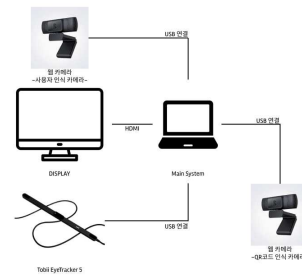


[그림 4] 키오스크 시스템의 작동 순서

본 시스템의 작동순서는 [그림 4]와 같다. 먼저 Web Camera를 통해 사용자를 인식하면(1)Main System에게 신호를 전송하여 키오스크 시스템을 동작시킨다.(2)Main System은 EyeTracker를 동작시켜(3) 사용자의 시선 위치에 대한 좌표값을 받는다.(4) 좌표 값은 Main System에게 전송하여 2초간 같은 위치를 응시 했다면 클릭하여 처리한다.(5) 멤버십 적립을 위해 Web Camera를 통해 QR코드를 인식하고,(6) 데이터 베이스를 통해 멤버십을 적립한다.(7)

### 3.1 H/W 구성

하드웨어 구성도는 [그림 5]와 같으며, 사용한 하드웨어 부품은 [표 2]와 같다. 사용자 얼굴 인식 기능의 Web Camera①에서 사람의 얼굴이 인식되면 키오스크가 동작한다. Eyetracker는 사용자의 시선 이동에 따른 좌표 값을 Main System에 전송하고, LCD Pannel을 통해 값에 따라 UI를 사용자에게 제공한다. 주문이 완료된 후 멤버십 적립을 하기 위한 QR코드를 제공하는데, 이를 인식하기 위해 Web Camera②가 동작한다.



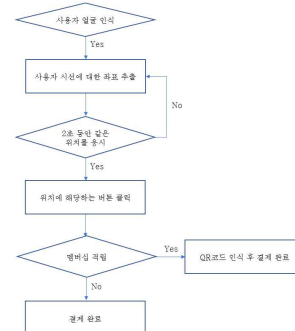
[그림 5] H/W 구성도

장비	기능
Web Camera①	사용자 얼굴 인식
LCD Pannel	키오스크 UI 출력
Eye Tracker	사용자 시선 인식
Web Camera②	QR 코드 인식
Main COM	키오스크 시스템 관리

[표 2] 하드웨어 부품 리스트

### 3.2 S/W 구성

본 시스템에서는 C++ 기반의 Tobii사의 Eye-Tracker를 통해 시선이 화면의 어디에 위치하는지 인식하고 이를 변수로 받아 마우스를 제어한다. 마우스 움직임, 클릭을 통해 키오스크 UI를 자유롭게 사용하고 QRCode 어플리케이션을 통해 사용자는 멤버십 포인트를 이용할 수 있다. 이는 Firebase 기반 데이터베이스에 저장된다.



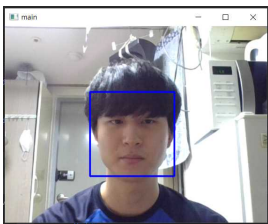
[그림 6] 키오스크 시스템 흐름도

키오스크 시스템의 흐름도는 [그림 6]과 같다. 사용자의

얼굴이 인식되었을 때, 사용자 시선에 대해 좌표를 추출하고 그 좌표를 2초 간 응시했다면 좌표에 해당하는 버튼에 대해 클릭 이벤트를 처리한다. 2초간 응시하지 않았다면 사용자 시선에 대한 좌표를 계속해서 추출하고 선택이 완료되면 멤버십 적립 여부에 따라 결제를 완료하거나 QR코드를 인식한 후 결제를 완료한다.

### 3.2.1 사용자 인식

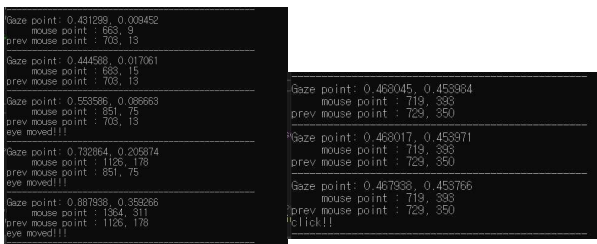
키오스크가 동작하기 위해 먼저 사용자의 얼굴을 인식한다. Python 기반의 Haar Cascade 알고리즘을 이용하여 [그림 7]과 같이 얼굴을 검출하며, 약 3초간 얼굴이 인식되었을 때 키오스크가 자동으로 동작한다.



[그림 7] 사용자 얼굴 인식

### 3.2.2 Eye Tracking

Tobii 사의 EyeTracker 제품을 사용하여 Eyetracking을 구현하였으며 본 Tobii사에서 제공되는 API를 통해 값을 통해 마우스를 제어한다. 화면 비율을 1로 했을 때, 보내진 Gaze point 값을 화면 비율에 맞추어 좌표로 전환해 마우스 움직임을 제어하며 [그림 8]과 같이 확인할 수 있다. 이전 좌표를 기준으로 50 이상의 차이가 있다면 다른 곳을 보고 있다고 판단하고 같은 곳을 2초 응시하면 해당 좌표를 클릭하고 [그림 9]와 같이 확인할 수 있다.



[그림 8] 시선 이동 인식 (좌)

[그림 9] 시선추적 값을 통한 마우스 제어 (우)

### 3.2.3 키오스크 UI

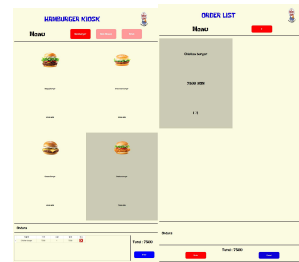
PyQt를 이용해 키오스크의 UI를 구축한다. PyQt는 C++의 Cross Platform GUI Framework인 Qt의 레이어아웃에 Python 코드를 연결하여 GUI 프로그램을 만들 수 있게 해

주는 Framework이다.

본 시스템의 UI는 메인화면, 주문목록 화면, QR 코드 인식화면 등으로 이루어져 있다.

메인 화면(Base Window / main\_window)은 [그림 10]과 같으며, 카테고리 선택 및 메뉴를 선택할 수 있는 화면이다. PyQt의 경우 메인에서 최초로 실행된 창이 종료될 시 다른 창의 종료 여부와 상관없이 프로그램이 종료되므로 메인화면(Base Window)은 결제가 완료되어도 초기 상태로 돌아갈 뿐 종료되지 않으며 계속해서 실행된다. 메인 화면 이후의 창들은 모두 메인화면에서 주문을 담는 Table Widget을 변수로 받아 그 값을 사용 하며, 이를 토대로 포인트 적립 및 영수증을 출력한다. 이 때문에 메인 화면을 Base Window로 둔다.

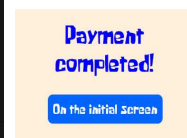
주문목록 화면(order\_window)은 [그림 11]과 같으며, 메인 화면의 Table Widget 목록을 크게 확인할 수 있다. 각 메뉴버튼들을 2초 이상 응시함으로써 테이블에 담긴 메뉴의 개수를 하나씩 차감하는 형식으로 구성되어 있다.



[그림 10] 키오스크 메인화면 (좌)

[그림 11] 주문 목록화면 (우)

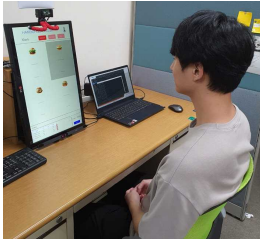
QR 코드 인식 화면(QRcode\_window)는 [그림 12]와 같으며, 전용 QR 코드 어플리케이션의 QR 코드를 카메라에 인식한다. 그 다음이미지 디코딩 및 데이터 전송의 과정을 거쳐 캡처 스크레드를 종료한다. 이후 Firebase를 기반으로 한 데이터베이스에 포인트를 적립한다. 결제가 완료되면 [그림 13]과 같이 결제가 완료되었다는 화면을 출력하고, 영수증을 출력한 후 초기화면으로 돌아간다.



[그림 12] QR 코드 인식 화면 (좌)

[그림 13] 결제 완료 화면 (우)

#### 4. 실험 및 평가



[그림 15] Eye-Tracking 성능 실험

[그림 15]와 같이 키오스크 시스템의 성능에 대한 실험을 진행했으며, ① 사용자 인식 실험 (사람의 얼굴이 인식되었을 때 키오스크 시스템의 동작 여부), ② Eye-Tracking 실험 (사용자의 시선에 따른 정확한 동작 여부)에 대해 실험했다.

##### ① 사용자 인식 실험

사용자 인식기능의 성능을 실험한 결과 거의 100%의 인식률을 이끌어 냈으나 마스크를 착용하고 있으면 인식이 불가능하다는 점을 도출했다.

##### ② Eye-Tracking 실험

	Eye-Tracking (%)				
	1	2	3	4	5
A	92	95	91	92	91
B	93	91	90	94	93
C	95	93	91	95	94
D	94	90	95	90	94
E	91	93	94	93	90

[표 3] Eye-Tracking 실험 결과

Eye-Tracking의 실험 결과는 [표 3]과 같으며 사용자가 시선을 이동하여 마우스 커서가 목표 까지 지정될 때 까지의 동작을 실험했다. 5명의 패널에 대해 5번의 실험을 진행했으며 사용자의 눈이 깜빡이는 것과는 관계 없이 시스템이 동작하는 것을 확인했다. 그 결과 패널마다 인식률에 미비한 차이가 있으나 평균 인식률 93%를 달성함으로써 키오스크 시스템이 원활하게 작동됨을 확인할 수 있었다.

#### 5. 결론

이 논문에서는 Haar Cascade 알고리즘을 이용한 사용자 얼굴 인식, Eyetracker를 이용한 Eye-Tracking 기술을 융합하여 직접적인 터치 없이 동작하는 키오스크 시스템을 구현해보았다. 사용자의 얼굴을 인식할 때 에러율이 현재 미비하지만, 최상의 인식률을 이끌어내기 위해서는 다양한 각도의 얼굴에 대한 데이터셋이 필요할 것으로 보인다.

더 나아가 거동이 불편한 사람들에게 키오스크 시스템을

자유롭게 사용할 수 있게 함으로써 노인이나 알츠하이머 환자, 파킨슨 병 환자 등이 최신 시스템을 활용할 수 있는 적절한 솔루션을 제공할 것으로 예상된다.

추가적으로 실시간 객체 탐지 알고리즘인 YOLO를 사용하여 실시간 검출을 통해 마스크 착용에 관한 문제를 해결하여 시스템을 고도화 할 수 있다.[6]

#### <참고문헌>

[1] 코로나19 실시간 상황판

<https://coronaboard.kr/>

[2] 포스단말기 및 무인주문기(키오스크) 사용여부

[https://kosis.kr/statHtml/statHtml.do?orgId=114&tblId=DT\\_114054\\_014](https://kosis.kr/statHtml/statHtml.do?orgId=114&tblId=DT_114054_014)

[3] eye tracking 원리

<https://www.tobii.com/ko/learn-and-support/learn/eye-tracking-essentials/how-do-tobii-eye-trackers-work/>

[4] Paul Viola, Michael Jones. Rapid 'Object Detection using a Boosted Cascade of Simple Features'

[5] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, Yui Man Lui 'Visual Object Tracking using Adaptive Correlation Filters'

[6] YOLO: Real-Time Object Detection

<https://pjreddie.com/darknet/yolo/>

※ 본 논문은 과학기술정보통신부

정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다.